



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/766,336	01/28/2004	Shyh-Mei F. Ho	SVL920030102/USI	3271
45112 7590 11/25/2008				
Kunzler & McKenzie 8 EAST BROADWAY SUITE 600 SALT LAKE CITY, UT 84111				
EXAMINER				
ALVESTIEFFER, STEPHEN D				
ART UNIT		PAPER NUMBER		
2175				
MAIL DATE		DELIVERY MODE		
11/25/2008		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/766,336

Applicant(s)

HO ET AL.

Examiner

Stephen Alvesteffer

Art Unit

2175

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 September 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 29,31-37,41 and 42 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 29,31-37,41 and 42 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SF/08)
Paper No(s)/Mail Date 20080910
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

Response to Amendment

This Office Action is responsive to the Request for Continued Examination (RCE) filed September 9, 2008. Claim 29 is currently amended. Claims 1-28, 30, and 38-40 are cancelled. Claim 29 is independent. Claims 29, 31-37, 41, and 42 remain pending.

The Information Disclosure Statement (IDS) filed September 10, 2008 has been considered by the examiner.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 29 and 31-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach et al. (hereinafter Bach '739), United States Patent 5,781,739, Bach et al. (hereinafter Bach '660), United States Patent 6,141,660, and Francis et al. (hereinafter Francis), United States Patent number 6,665,861.

Regarding claim 29, Bach '739 substantially teaches a method for automatically generating a web interface for an MFS-based IMS application, comprising:

an interface accepting a parameter set provided as a single input (Bach '660 addressed below);

importing MFS-based IMS source files corresponding to an MFS-based IMS application (see Bach '739 column 2 lines 62-66; "...browse and download MFS source information");

generating at least one eXtended Markup Language Metadata Interchange (XMI) file for the MFS-based IMS application associated with the imported MFS-based IMS source files (Francis, addressed below); and

generating a middleware application from the at least one XMI file, the middleware application configured to interface between a client application and the MFS-based IMS application (see Bach '739 column 10 lines 44-49; "*The generated CGI-BIN program invokes this class to parse the input string from the Web browser*", code is generated to interface between a web browser and the IMS database); and

automatically deploying without human intervention the generated middleware application to one or more servers (Bach '660, addressed below).

Bach '739 does not teach an interface accepting a parameter set provided as a single input. Configuring an interface to execute several different instruction sets in response to a parameter set provided as a single input was well known in the art at the time the invention was made. Bach '660 teaches using batch processing to execute several functions at once in response to a parameter set provided as a single input to the interface (see Bach '660 column 17 lines 54-67; "*The RUNSCRIPT command runs a file that contains all of the necessary commands. A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT*

command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403. In theory, this should save them a lot of time (the user may not want to go through all the GUI 402 panels for each database if they have a substantial number of databases). Users can also concatenate several script files into one and just run that single file through the CLI.").

Bach '739 further does not explicitly teach automatically deploying without human intervention the generated middleware application to one or more servers. Beginning in column 16 line 63, Bach '660 details the operation of the IMS object connector class command line interface. Using the command line interface of Bach '660, the user may enter all the commands for generating and deploying the middleware in a single batch file (see Bach '660 column 17 lines 4-12; *"Using the CLI 403, the user can enter commands one by one, or can run a command script that contains all of the commands"*). The user may put a GENERATE command in the batch file to generate the middleware (see Bach '660 column 19 lines 64-67; *"The GENERATE command creates the classes based on the information that is specified in the DEFINECLASS command"*). Finally, there is an UPLOAD command that can be used to automatically deploy any file to any server (see Bach '660 column 17 lines 42-48; *"The commands for downloading and uploading files from the server computer 102 (LOGIN, CD, GETFILE, UPLOAD, and DISCONNECT) can be entered at any time. However, CD, GETFILE, and DISCONNECT commands can be entered only after logging onto the server computer 102"*). Also of note is the TARGET attribute of the GENERATE command,

which can deploy generated classes to any server (see Bach '660 column 20 lines 6-12; *"directory is the name of the directory where the generated C++ classes are to be stored"*). Bach '660 teaches automatic deployment of generated middleware without human intervention.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide the batch and command line interface of Bach '660 in the invention of Bach '739 for the purpose of saving time and automating tedious multiple-command entry on an interface (see Bach '660 column 17 lines 53-67; *"A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403. In theory, this should save them a lot of time (the user may not want to go through all the GUI 402 panels for each database if they have a substantial number of databases)."*).

Neither Bach '739 nor Bach '660 teaches generating at least one eXtended Markup Language Metadata Interchange (XMI) file for the MFS-based IMS application associated with the imported MFS-based IMS source files. Francis teaches exchanging data between web applications and database applications using XMI (see Francis column 6 line 59 through column 7 line 22; *"The XML Metadata Interchange Format (XMI) specifies an open information interchange model that is intended to give developers working with object technology the ability to exchange programming data*

over the Internet in a standardized way, thus bringing consistency and compatibility to applications created in collaborative environments"). XMI was a well-known standard for exchanging data at the time the invention of the instant application was made. Both Bach '739 and Bach '660 solve the problem of accessing data on legacy systems (see Bach '739 Appendix A and Bach '660 column 4 lines 2-10). It would have been obvious to one of ordinary skill in the art at the time the invention was made, having the disclosures of Bach '739/'660 and Francis laid before him, to use XMI as a common format for exchanging data (as taught by Francis) between the source and target databases of Bach '739/'660 in order to standardize interaction between different systems (see Francis column 6 line 59 through column 7 line 22).

Regarding claim 31, Bach '739/Bach '660/Francis teaches loading a script comprising the parameter set from persistent storage (see Bach '660 column 17 lines 4-12; *"the user can enter commands one by one, or can run a command script that contains all of the commands"*).

Regarding claim 32, Bach '739/Bach '660/Francis teaches that the script comprises a plurality of parameter sets each associated with a different MFS-based IMS application (see Bach '660 column 17 lines 4-12; *"the user can enter commands one by one, or can run a command script that contains all of the commands"*, the command line interface described by Bach '660 is capable of comprising a plurality of parameter sets each associated with a different MFS-based IMS application).

Regarding claim 33, Bach '739/Bach '660/Francis teaches that the parameter set is manually entered, the method further comprising storing the manually entered

parameter set (see Bach '660 column 17 lines 54-67; "A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403").

Claims 34, 35, 41, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, Francis (6,665,861) *supra*, and Womble, United States Patent 5,488,648.

Regarding claim 34, Bach '739/Bach '660/Francis teaches every limitation of claim 34, but does not explicitly disclose that the interface comprises a plurality of modes, each mode involving a different level of user interaction. However, it was well known in the art at the time the invention was made that some, if not most, command-line programs allow users to enter no parameters, some parameters, or all parameters to operate the program, as evidenced by Womble (see Womble column 13 lines 33-43; "This portion of the LOGEVENT program is invoked from the DOS prompt. If no parameters are supplied, a menu system will prompt the user for the input to the program. Command line parameter switches can appear anywhere on the command line as long as they are separated by spaces. Switches consists of a hyphen, a letter and a plus or minus sign to turn the switch on or off (some switches are followed by a string instead of the plus and minus sign) and most switches have a default setting. It is

only necessary to supply a switch if the default setting is not satisfactory"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide command line interface modes accepting none, some, or all possible parameters as taught by Womble in the code generating interface as taught by Bach '739/Bach '660/Francis in order to provide different levels of interaction with the invention.

Regarding claim 35, Bach '739/Bach '660/Francis/Womble teaches that one mode comprises a batch mode that reads the parameter set from persistent storage (see Bach '660 column 17 lines 54-67; *"The RUNSCRIPT command runs a file that contains all of the necessary commands. A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403. In theory, this should save them a lot of time (the user may not want to go through all the GUI 402 panels for each database if they have a substantial number of databases). Users can also concatenate several script files into one and just run that single file through the CLI."*).

Regarding claim 41, Bach '739/Bach '660/Francis/Womble teaches that one mode comprises a novice mode that prompts a user for each parameter in the parameter set (see Womble column 13 lines 33-43; *"Command line parameter switches can appear anywhere on the command line as long as they are separated by spaces.*

Switches consists of a hyphen, a letter and a plus or minus sign to turn the switch on or off (some switches are followed by a string instead of the plus and minus sign) and most switches have a default setting. It is only necessary to supply a switch if the default setting is not satisfactory").

Regarding claim 42, Bach '739/Bach '660/Francis/Womble teaches that one mode comprises an expert mode that prompts a user to enter the parameter set as a single entry (see Womble column 13 lines 33-43; *"If no parameters are supplied, a menu system will prompt the user for the input to the program"*).

Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, Francis (6,665,861) *supra*, and Dan et al. (hereinafter Dan), United States Patent number 6,560,639.

Regarding claim 36, Bach '739/660/Francis explicitly teaches every limitation of claim 36 except presenting an error message in response to an error condition triggered when generating the at least one XMI file. However, including an error message to respond to application errors was well known in the art at the time the invention was made, as evidenced by Dan (see Dan column 5 lines 1-11; *"error report manager may report any error in intended user changes to a requested web page"*). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include an error module as described by Dan in the invention of Bach '739/Bach '660/Francis for the purpose of handling error conditions in the application.

Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, Francis (6,665,861) *supra*, and Snover et al. (hereinafter Snover), United States Patent Application Publication 2004/0230987.

Regarding claim 37, Bach '739/Bach '660/Francis teaches every limitation of claim 37 except that importing comprises importing a plurality of MFS-based IMS source files from a single directory in response to a single parameter. However, it was well known in the art at the time the invention was made that a wildcard character can be entered on many command line interfaces to specify several files using a single parameter, as indicated by Snover (see Snover paragraph [0054]; *"the service could perform wildcard expansion on a filename entered as "A*" on the command line. Before the second pass, the Name member contains "A*" as it was entered on the command line. During the second pass, the Filename service may locate a set of files starting with A and store them in the Arraylist F"*). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the wildcard expansion technique of Snover with the web interface generator of Bach '739/Bach '660/Francis in order to make importing several files at the command line less tedious.

Response to Arguments

Applicant asserts that none of the cited references teach automatic deployment of the generated XMI files and middleware applications. The examiner respectfully disagrees.

Beginning in column 16 line 63, Bach '660 details the operation of the IMS object connector class command line interface. Using the command line interface of Bach '660, the user may enter all the commands for generating and deploying the middleware in a single batch file (see Bach '660 column 17 lines 4-12; *"Using the CLI 403, the user can enter commands one by one, or can run a command script that contains all of the commands"*). The user may put a GENERATE command in the batch file to generate the middleware (see Bach '660 column 19 lines 64-67; *"The GENERATE command creates the classes based on the information that is specified in the DEFINECLASS command"*). Finally, there is an UPLOAD command that can be used to automatically deploy any file to any server (see Bach '660 column 17 lines 42-48; *"The commands for downloading and uploading files from the server computer 102 (LOGIN, CD, GETFILE, UPLOAD, and DISCONNECT) can be entered at any time. However, CD, GETFILE, and DISCONNECT commands can be entered only after logging onto the server computer 102"*). Also of note is the TARGET attribute of the GENERATE command, which can deploy generated classes to any server (see Bach '660 column 20 lines 6-12; *"directory is the name of the directory where the generated C++ classes are to be stored"*). Bach '660 teaches automatic deployment of generated middleware without human intervention.

Applicant asserts that there is no evidence that one of skill in the art would have found it obvious to use XML in approaching the problems that gave rise to the instant application. The examiner respectfully disagrees.

Francis teaches exchanging data between web applications and database applications using XML (see Francis column 6 line 59 through column 7 line 22; "*The XML Metadata Interchange Format (XML) specifies an open information interchange model that is intended to give developers working with object technology the ability to exchange programming data over the Internet in a standardized way, thus bringing consistency and compatibility to applications created in collaborative environments*"). XML was a well-known standard for exchanging data at the time the invention of the instant application was made. Both Bach '739 and Bach '660 solve the problem of accessing data on legacy systems using newer systems, such as web-based systems (see Bach '739 Appendix A and Bach '660 column 4 lines 2-10). It would have been obvious to one of ordinary skill in the art at the time the invention was made, having the disclosures of Bach '739/'660 and Francis laid before him, to use XML as a common format for exchanging data (as taught by Francis) between the source and target databases of Bach '739/'660 in order to standardize interaction between different systems (see Francis column 6 line 59 through column 7 line 22).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Stephen Alvesteffer whose telephone number is (571)270-1295. The examiner can normally be reached on Monday-Friday 9:30AM-6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Bashore can be reached on (571)272-4088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Stephen Alvesteffer
Examiner
Art Unit 2175

/S. A./
Examiner, Art Unit 2175

/Kieu D Vu/
Primary Examiner, Art Unit 2175